



**University of  
Zurich**<sup>UZH</sup>

**Zurich Open Repository and  
Archive**

University of Zurich  
University Library  
Strickhofstrasse 39  
CH-8057 Zurich  
[www.zora.uzh.ch](http://www.zora.uzh.ch)

---

Year: 2002

---

## **Personalisierbare elektronische Gruppenarbeitsräume**

Schwabe, Gerhard ; Löber, Andreas

Posted at the Zurich Open Repository and Archive, University of Zurich

ZORA URL: <https://doi.org/10.5167/uzh-67790>

Conference or Workshop Item

Originally published at:

Schwabe, Gerhard; Löber, Andreas (2002). Personalisierbare elektronische Gruppenarbeitsräume. In: Herzeg, M.; Oberquelle, H.; Prinz, W.: Mensch und Computer, unknown, 1 January 2002.

# Personalisierbare elektronische Gruppenarbeitsräume

Gerhard Schwabe, Andreas Löber, Volker Riediger, Universität Koblenz-Landau

## Zusammenfassung

Mit dem Aufkommen neuer Endgeräte wächst das Interesse an elektronischen Gruppenarbeitsräumen. Die schon lange erhobenen Anforderungen nach sehr großen stationären und sehr kleinen mobilen Endgeräten sind mit der Verbreitung von Beamern und Plasmabildschirmen für elektronische Tafeln sowie funkvernetzten Notebooks und PDAs jetzt mit kommerziell verfügbarer Technologie im Prinzip erfüllbar. Für die verteilte Zusammenarbeit steht erprobte Technologie zum Videoconferencing zur Verfügung und ein reichhaltiges Angebot an Software unterstützt die „Any Place-Any Time“ Zusammenarbeit.

Und doch entstehen gerade durch diese neue Technik weitere Hindernisse. Um das komplexe und schwer bedienbare Gefüge der vorhandenen Technik effektiv nutzen zu können, bedarf es neuer Software, die den Moderator aktiv bei seiner Arbeit unterstützt, und eine leichte Verwendung interaktiver Räume ermöglicht. Mit NOAH entstand an der Universität Koblenz-Landau eine solche Software. NOAH ermöglicht es dem Moderator, sich die elektronische Einrichtung seines Raums auf seine Bedürfnisse zuzuschneiden und diese Einstellungen auch über das Ende einer Sitzung hinaus für eine Folgesitzung oder als Default für eine spätere Sitzung zu speichern. Dieser Beitrag motiviert die Notwendigkeit einer solchen Unterstützung, stellt NOAH und den elektronischen Gruppenarbeitsraum vor und gibt einen Überblick auf die Architektur von NOAH.

## 1 Einführung und Problemstellung

Mit dem Aufkommen neuer Endgeräte wächst das Interesse an elektronischen Gruppenarbeitsräumen. Die schon zu Beginn der 90er Jahre erhobenen Anforderungen (vgl. Weiser 1991) nach sehr großen stationären und sehr kleinen mobilen Endgeräten sind mit der Verbreitung von Beamern und Plasmabildschirmen für elektronische Tafeln sowie funkvernetzten Notebooks und PDAs jetzt mit kommerziell verfügbarer Technologie im Prinzip erfüllbar. Für die verteilte Zusammenarbeit steht erprobte Technologie zum Videoconferencing zur Verfügung und ein reichhaltiges Angebot an Software unterstützt die „Any Place-Any Time“ Zusammenarbeit. Dennoch haben sich insbesondere anspruchsvollere Anwendungen noch nicht durchgesetzt (vgl. Schwabe 2002). Ein Grund hierfür ist die Komplexität der Gesamtsysteme für den Anwender. Die Vorbereitung, Einrichtung und Bedienung der elektronischen Arbeitsumgebung erfordert so viele Vorkenntnisse, dass weite Anwenderkreise ausgeschlossen werden, und so viel Zeit, dass viele Gelegenheiten für computerunterstützte Gruppenarbeit ausgelassen oder konventionell abgewickelt werden. Elektronische Gruppenarbeitsräume werden sich also nur dann weiter verbreiten, wenn sie einfacher zu verwenden sind. Hierzu gehören die Möglichkeiten

- Standardsitzungen ohne umfangreiche Vorbereitungsarbeiten zu beginnen und einfach durchzuführen,
- Gruppenarbeitsprozesse zu unterbrechen und später (auch Tage später) an dem Punkt fortzusetzen, wo sie unterbrochen wurden,

- Gruppenarbeitsprozesse nicht nur inhaltlich, sondern auch von der eingesetzten Hard- und Software her einfach vorzubereiten; hierzu gehört auch, Vorbereitungsarbeiten und die eigentliche Sitzungszeit zeitlich zu entzerren
- Neue Sitzungen aus bewährten Bausteinen vergangener Sitzungen zusammenzusetzen, anstatt sie jeweils von Grundauf neu zu beginnen.

Diese wünschenswerten Eigenschaften eines elektronischen Gruppenarbeitsraums dürfen aber die Möglichkeit des verantwortlichen Moderators nicht beschränken, während der Sitzung flexibel die Arbeitsumgebung an die Bedürfnisse der Gruppe anzupassen.

Diese funktionalen Anforderungen führen zu folgenden technischen Anforderungen an elektronische Gruppenarbeitsräume:

- Die für größere Gruppen erforderliche zentrale Steuerung muss alle Komponenten verwalten, d.h. es müssen Hardware und Softwareanwendungen von einem Schaltpult aus gesteuert werden
- Die Einstellungen des Systems müssen persistent sein können. Dies gilt wiederum sowohl für Hardware als auch für Software.
- Da ein Gruppenarbeitsraum von verschiedenen Gruppen genutzt wird, müssen die Einstellungen des Raums einzelnen Gruppen bzw. Moderatoren zugeordnet werden. Die Raumeinstellung muss somit personalisiert werden.
- Die Raumausstattung muss dem technischen Fortschritt angepasst werden können; insbesondere müssen Komponenten ausgewechselt werden können. Dies erfordert eine flexible Systemarchitektur und die weitgehende Abstraktion von konkreten technischen Komponenten.

Dieser Beitrag zeigt am Beispiel des Noah-Systems, wie ein elektronischer Gruppenarbeitsraum durch eine geeignete Software einfacher nutzbar wird. Hierzu wird zuerst im Kapitel 2 der elektronische Gruppenarbeitsraum und die Systemfunktionalität vorgestellt. Kapitel 3 beschreibt die Systemarchitektur von Noah. Der Beitrag schließt mit Schlussfolgerungen und einem Ausblick sowie einer kurzen Diskussion verwandter Arbeiten.

## 2 Systemfunktionalität aus Benutzersicht

Die Systemfunktionalität von Noah (Network Oriented Agenda Helper) sei in Anlehnung an Fallstudienworkshops zum computerunterstützten Lernen erläutert<sup>1</sup>. Aufgabe der 10 Studierenden soll es sein, im Laufe des Semesters insgesamt sechs Fallstudien aus dem Bereich der Wirtschaftsinformatik in einem Wechsel aus Einzelarbeit, Kleingruppenarbeit und Plenumsarbeit zu lösen. Für jede Fallstudie sind insgesamt zwei Wochen angesetzt. Nach einer Einarbeitung in die Fallstudie treffen sich die Studierenden in der ersten Woche zu einem vierstündigen Workshop, in dem sie in Kleingruppenarbeit und Plenumsarbeit die Schlüsselprobleme des Falles erarbeiten und Ideen für Lösungsvorschläge sammeln. Jede Kleingruppe erhält die Aufgabe, innerhalb einer Woche die Problemanalyse zu vertiefen und mit Hilfe der gesammelten Ideen und eigener Recherchen ein

---

<sup>1</sup> Diese computerunterstützten Fallstudienworkshops wurden in ähnlicher Weise in Koblenz im Wintersemester 2001/2002 veranstaltet, allerdings wurde nicht im beschriebenen Umfang die Technologie eingesetzt

Lösungskonzept zu erarbeiten. Die Lösungskonzepte werden dann in der Folgewoche präsentiert, diskutiert und einander gegenübergestellt.

Vor Beginn der Lehrveranstaltung richtet Dozent Schütterle einmal das Koblenzer CSCW-Labor (vgl. Abbildung 1) für seine Zwecke ein.



*Abbildung 1: Koblenzer CSCW-Labor*

Der große, auflichtprojizierte Bildschirm dient als zentrale Präsentationsfläche; die drei kleineren Plasmabildschirme dienen zur Präsentation von Zwischenresultaten und für die Kleingruppenarbeit. Jeder Student erhält einen Notebook oder einen Stiftcomputer für die individuelle Arbeit und einfache Kleingruppenarbeit. Da der Raum zu klein für einzelne Kleingruppenaktivitäten ist, soll eine Videoverbindung zu einem zweiten Arbeitsraum in einzelnen Phasen auf einer elektronischen Tafel aufgebaut werden. Über diese Videoverbindung kann sich Dozent Schütterle auch seinem Assistenten Häberle, der die Studierenden im zweiten Raum betreut, abstimmen. Für die Lehrveranstaltung bereitet Häberle mit Noah eine Standardtagesordnung vor, die er dann als Schablone für weitere Tagesordnungen verwendet (vgl. Abbildung 22).

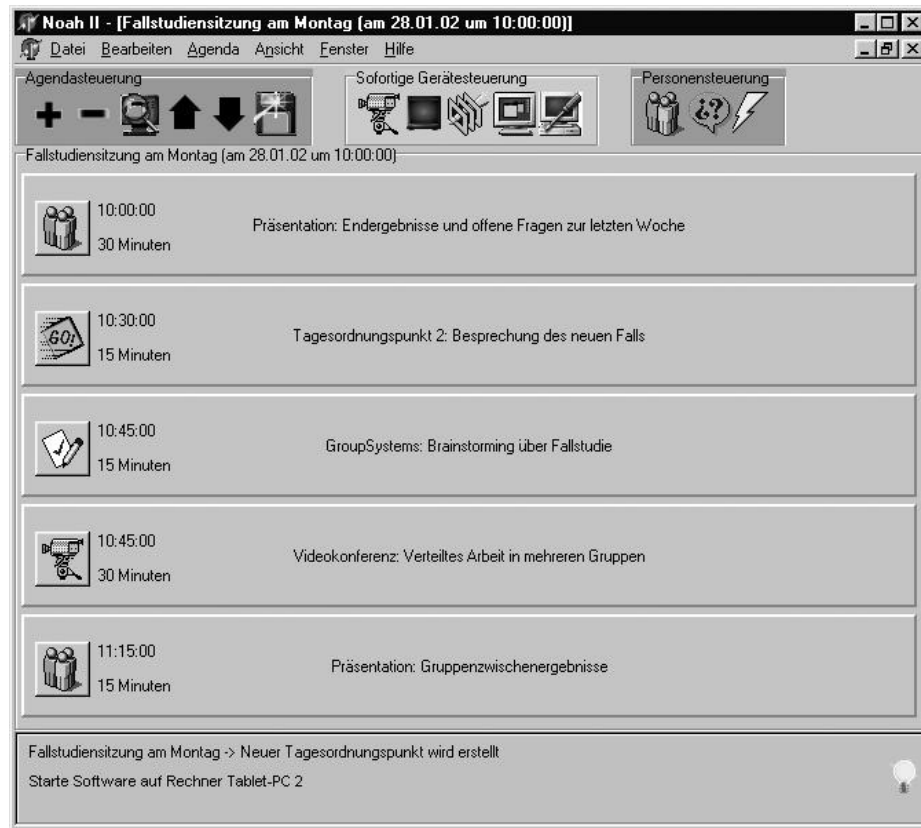


Abbildung 2: Das Noah-Tagesordnungswerkzeug

Die Tagesordnung beschreibt die einzelnen Aktivitäten. Für jede Aktivität wird eine eigene Raumkonfiguration vorgesehen. Eine Raumkonfiguration legt fest:

- Welche Personen an welchen Rechnern teilnehmen
- Welches Werkzeug welche Person verwendet – dies kann beispielsweise ein Präsentationstool, eine Standardtextverarbeitung oder die Moderationssoftware GroupSystems sein
- Mit welchen Dokumenten die einzelnen Werkzeuge gestartet werden sollen
- Welche Informationen auf den elektronischen Tafeln zu sehen sind
- Welche Videokonferenzverbindung aufgebaut werden soll, auf welcher elektronischen Tafel sie angezeigt werden soll und wie die Kameras und Mikrophone eingestellt werden sollen.

Für die einzelnen Einstellungen verwendet man Spezialwerkzeuge, die vom Tagesordnungswerkzeug aus gestartet werden können. Ein Beispiel für ein solches Spezialwerkzeug ist die Kamerasteuerung (vgl. Abbildung 3), mit dem den einzelnen Kameras (C1-C3) eine aufzunehmende Region zugeordnet wird und Kamerainput aus anderen Räumen den Displays zugeordnet werden. Aufbauend auf dieser ersten Default-Tagesordnung erstellt Schütterle drei Varianten: Eine erste Variante der Tagesordnung ist für die intensive Gruppenarbeit in der ersten Woche der Fallbear-

beitung, eine zweite Variante für die Präsentationen in der zweiten Woche und eine dritte Variante stellt der den einzelnen Studenten für ihre Kleingruppenarbeit zwischen den Terminen zur Verfügung.

Zu Beginn einer jeden Lehrveranstaltung stellt er durch Auswahl einer Tagesordnung die Ausgangskonfiguration für seine Lehrveranstaltung ein – unabhängig davon, wie die Raumeinstellung während der Woche verändert wurde. Während der einzelnen Tagesordnungspunkte kann er gezielt die vorbereiteten Konfigurationen verwenden (indem er z.B. Werkzeuge an den einzelnen Arbeitsplätzen startet) oder sie mit einer Option zur sofortigen Gerätesteuerung modifizieren.

Der Raum ist für die Zeit seiner Nutzung auf Herrn Schütterle zugeschnitten – auch über längere Nutzungszeiten hinweg. Der Zwang zu einer zentralen Sitzungssteuerung ist in diesem Szenario noch vorhanden und widerspricht dem Ideal dezentral gesteuerten Lernens; es ist aber eine allgemeine Moderationserfahrung, dass größere Gruppen in Workshops diese Steuerung des Sitzungsprozesses und der verwendeten Werkzeuge (!) benötigen, um sich ungestört Inhalten widmen zu können<sup>2</sup>. Die technische Komplexität der Nutzung des Raumes reduziert sich aber deutlich im Vergleich zur Einzelsteuerung der jeweiligen Systemkomponenten.

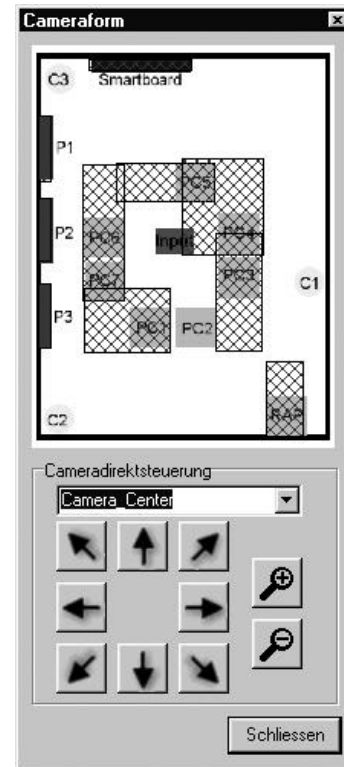


Abbildung 3: Kamerasteuerung

### 3 Architektur von Noah

Die Architektur von Noah sei an seinem Objektmodell sowie an einem Schichtenmodell für die Implementierung vorgestellt (vgl. Abbildung 4). Ziel der Gestaltung von Noah war es, den Nutzer möglichst weitgehend von den technischen Spezifika der einzelnen Geräte zu entlasten. Ein Moderator möchte beispielsweise die Kamera nach rechts drehen; dazu interessieren ihn weder die Spezifika der Kamerabefehle von Sony, noch ob er diesen Befehl über eine serielles Kabel oder über getunneltes IP weiterleiten muss. Der Wechsel der Raumkonfiguration für eine neue Teilaufgabe innerhalb einer Gruppensitzung erfordert die Änderung bzw. Einstellung von teilweise mehr als 50 Geräteparametern. Ohne eine Systemunterstützung wäre eine Nutzung des Raums daher völlig unpraktikabel.

<sup>2</sup> Dies gilt z.B. auch für konventionelle Moderationsmethoden und -materialien wie Metaplan.

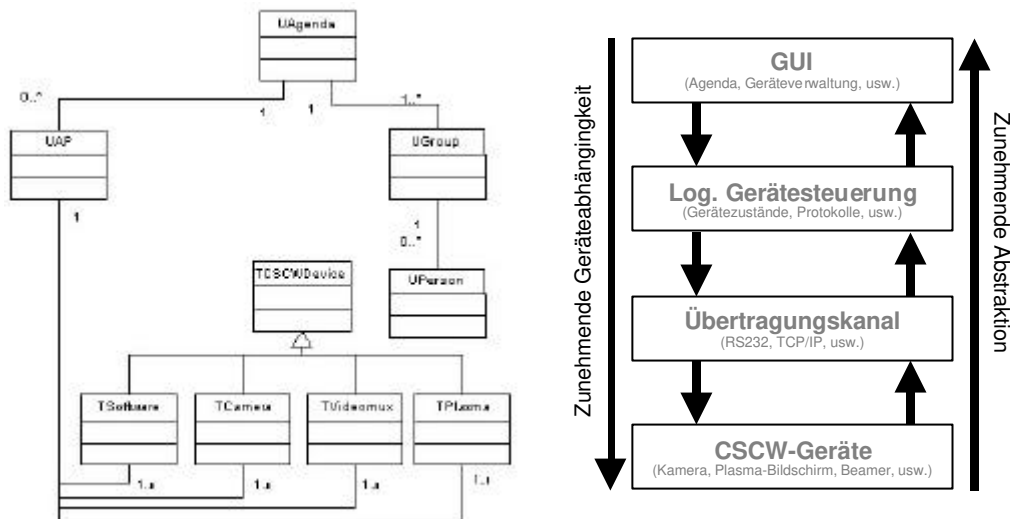


Abbildung 4: Architektur: Klassenmodell und Schichtenmodell

Kern des Modells ist die abstrakte Klasse *CSCWDevice*. In ihr werden Funktionen verwaltet, die für alle Geräte einheitlich sind. Ein Gerät kann ein *Projektionsdevice*, ein *Videomix*, eine *Camera* oder *Software* sein. Ein *Projektionsdevice* ist wiederum eine abstrakte Klasse unter der dann die konkreten Gerätetypen zugeordnet sind (z.B. ein *Plasmabildschirm*, ein *Beamer*...; angesteuert wird jeweils eine Einheit aus Endgerät plus PC). *Videomix* kapselt die Steuerung von programmierbaren Videomischern, die eine Anzahl von Videoeingängen über eine Kreuzschiene mit verschiedenen Videoausgängen verbinden können. *Camera* stellt eine prototypische Kamera zur Verfügung, während die *Software*-Klasse die Möglichkeit zum zentral verwalteten Softwarestart auf den Teilnehmerrechnern bietet. Ferner bietet sich die Möglichkeit, durch spezielle Programme einen Teilnehmer-Rechner herunterzufahren, oder seine Softwarekonfiguration anzupassen. Die einzelnen Devices werden einzelnen Tagesordnungspunkten (*Agendapoint*) zugeordnet. Die Tagesordnungspunkte ergeben zusammen die Tagesordnung (*Agenda*). In der Tagesordnung werden außerdem die einzelnen (Teil-) Gruppen (*Group*) verwaltet, die in der Sitzung aktiv sind. Die Gruppen setzen sich wiederum aus einer Reihe von Personen (*Person*) zusammen.

Die in den genannten Klassen erfassten Daten werden periodisch in einer Datenbank gespeichert. Dadurch kann die Einstellung der Geräte jederzeit auf den letzten abgespeicherten Stand zurückgesetzt werden. Dies ist nicht nur bei technisch bedingten Unterbrechungen zweckmäßig, sondern auch, wenn sich die Kooperation wie im eingangs geschilderten Beispiel über längere Zeit hinzieht. Die gespeicherten Informationen sind jeweils spezifisch für eine Tagesordnung, d.h. eine neue Tagesordnung überschreibt die Einstellung einer alten Tagesordnung nicht.

Während das Klassenmodell die Abstraktion von konkreten Geräten darstellt, läßt sich die Abstraktion von konkreten technischen Steuerungsmechanismen am besten an einem Schichtenmodell darstellen (Abbildung 4 rechts): Ein Steuerbefehl wird in einem Hochsprachenbefehl (z.B. „Move camera left“) von dem Tagesordnungswerkzeug (*GUI*) an die *logische Gerätesteuerung* weitergegeben. Diese logische Gerätesteuerung übersetzt diese abstrakten Steuerbefehle in geräte-

spezifische Befehle. Die dafür benötigten Informationen erhält sie aus den in der Datenbank gespeicherten Klassenbeschreibungen. Die *logische Gerätesteuerung* schreibt die gerätespezifischen Befehle dann in einen Puffer. Von dort werden die Befehle vom *Übertragungskanal* abgeholt und zu einem Empfänger übertragen. Der abstrakte *Übertragungskanal* kann die Übertragung über verschiedene Protokolle verwalten, insbesondere entscheidet er, ob ein Gerät direkt über die serielle Schnittstelle angesprochen wird, oder der Befehl über IP getunnelt an das Gerät übertragen wird. Die vom *Übertragungskanal* in Pakete gepackten und weitergeleiteten Befehle können dann direkt vom *CSCW-Gerät* angenommen und umgesetzt werden. Die Rückmeldungen der Geräte werden dann entsprechend an das GUI zurückgemeldet.

Die Architektur erlaubt für Benutzung und Administration ein hohes Maß an Flexibilität: So kann die Vorbereitung einer Tagesordnung auch „Offline“ erfolgen. „Offline“ heißt hier, dass zeitgleich mit einer laufenden Sitzung auf einem beliebigen anderen Arbeitsplatz, beispielsweise im Büro des Moderators, eine neue Tagesordnung aufgestellt werden kann. Dies erfordert für den Bearbeiter natürlich eine gewisse Vertrautheit mit den örtlichen Gegebenheiten, erlaubt aber gleichzeitig eine hohe zeitliche Auslastung des CSCW-Raumes, weil keine zusätzlichen unproduktiven Belegungszeiten für die Sitzungsplanung vorzusehen sind.

Die einzelnen Projektionsflächen und Plasma-Displays sowie die Arbeitsplätze verfügen bereits über einen eigenen PC, der zur Ausführung und Anzeige der Applikationen benötigt wird. Dies erlaubt eine räumliche Trennung von gesteuerten Geräten und Regieplatz; damit ist jedes Gerät über die IP-Adresse des zugeordneten Computers erreichbar. Dies ermöglicht verteilte Sitzungen an mehreren Orten, wobei auch die Komponenten und Arbeitsplätze der entfernten Räume zentral gesteuert werden können.

Die vorgestellte Noah-Architektur ermöglicht es ohne weiteres, weitere Geräte eines bereits bekannten Typs hinzuzufügen, beispielsweise einen neuen Plasmabildschirm. Es ist auch vergleichsweise einfach, neue Gerätetypen hinzuzufügen, denn es muss nur eine neue Geräteklasse mit dem benötigten Protokoll angelegt werden und der Datenbank bekanntgegeben werden.

Die Architektur wurde mit Rücksicht auf direkte Implementierbarkeit gestaltet. Die konkrete Implementation in Delphi ist aber insbesondere wegen der Vielzahl der zu verwaltenden technischen Parameter, der Komplexität der individuellen Geräteprotokolle und der hohen Anforderungen an die Ergonomie des Interfaces sehr umfangreich. Insgesamt umfasst das Noah-System mehr als 100 Klassen und ca. 12.000 Lines of Code (ohne automatisch generierten Code).

Während die Steuerung von Projektionsdevices, Kameras und Videomux ohne größere Einschränkung umgesetzt werden konnte, ist die zentrale Steuerung von Applikationssoftware nur mit Einschränkungen möglich. Diese Einschränkungen ergeben sich zum einen aus den Parametrisierungsmöglichkeiten beim Programmaufruf und zum anderen durch die beschränkten Rückmeldungen des Betriebssystems (Windows NT) über den Erfolg oder Misserfolg eines entfernten Programmstarts. So ist es mit Noah ohne weiteres möglich, an jedem Arbeitsplatz Powerpoint mit einer ausgewählten Präsentation aufzurufen. Sollte Powerpoint aber abstürzen, dann gibt das Betriebssystem hierfür keine ausreichende Meldung weiter, so dass dieser Zustand der Steuerung nicht bekannt wird. Die Probleme begrenzter Parametrisierbarkeit von Softwareaufrufen lassen sich gut am Beispiel der Moderationssoftware GroupSystems zeigen: GroupSystems kann nur als Ganzes aufgerufen werden, nicht hingegen ein einzelnes Werkzeug von GroupSystems. GroupSystems hat ein eigenes Tagesordnungswerkzeug, eine eigene Gruppenverwaltung und eine eigene Werkzeugsteuerung. Die Schnittstellen von GroupSystems sind so schlecht dokumentiert, dass es auch nicht möglich war, die Funktionalität von Noah in die GroupSystems-Agenda einzubauen. Das Noah-Tagesordnungswerkzeug harmonisiert zwar gut mit GroupSystems, aber es bleibt



dem Moderator dennoch nicht erspart, die Gruppe über zwei verschiedene Tagesordnungswerkzeuge zu steuern.

## 4 Schlussfolgerungen und Ausblick

Eine wesentliche Erkenntnis der Ergonomieforschung ist es, dass es schwierig und aufwändig ist, einfach zu benutzende Software zu entwerfen. Dies hat sich hier auch bei der Vereinfachung der Benutzungsschnittstelle eines elektronischen Gruppenarbeitsraums gezeigt. In der jetzigen Fassung ermöglicht Noah, einen personalisierten Gruppenarbeitsraum für den Moderator und für die gesamte durch ihn moderierte Gruppe zu konfigurieren. Die Berücksichtigung der individuellen Präferenzen der Gruppenmitglieder beschränken sich auf die von ihnen im Laufe der Gruppenarbeit erarbeiteten Zwischenergebnisse, denn nur diese unterliegen ihrem direkten Einfluss. Dies lässt sich in dem derzeitigen Nutzungsszenario mit zentral verwalteten und zum Teil fest eingebauten Computern noch mit den Anforderungen der Moderation (d.h. der zentralen Prozesssteuerung) begründen. Mit der zunehmenden Verbreitung mobiler Endgeräte (Notebooks, PDAs) ist davon auszugehen, dass die Teilnehmer ihre eigene Technologie in den Raum mitbringen und über ein Funknetzwerk anbinden. Beispielsweise bringen die Studierenden des Fachbereichs Informatik der Universität Koblenz-Landau mit zunehmender Tendenz ihre eigenen funkvernetzten Notebooks mit in die Lehrveranstaltungen. In diesem Fall erfordert ein einfach zu verwendender elektronischer Gruppenarbeitsraum zumindest, dass die anwesenden Geräte automatisch erkannt und in die zentrale Prozesssteuerung eingebunden werden. Dies erfordert eine zusätzliche Klasse Geräteklasse *Sensor* (die durchaus in die Wireless Lan Funktionalität integriert sein kann). Persönliche Endgeräte können und sollen aber in geringerem Maße der Kontrolle eines Moderators unterliegen, als dies bei Geräten der Fall ist, die fest zu dem elektronischen Gruppenarbeitsraum gehören. Es ist eine besondere Herausforderung hier adäquate Kompromisse zwischen chaotischen Gruppenprozessen und einer unzumutbaren Überwachung zu finden. Wie weit sollte beispielsweise die Kontrolle des Dozenten über persönliche Endgeräte bei der Gruppenarbeit im Unterricht gehen und was soll der Student selbst entscheiden dürfen? Andererseits liegt gerade in elektronischen Gruppenarbeitsräumen als Rahmeninfrastruktur für persönliche Endgeräte ein großes Potential für eine effiziente Nutzung teurer Ressourcen. In einer Vision könnte man sich vorstellen, dass eine gewünschte Raumkonfiguration nicht mehr nur an einen konkreten physischen Arbeitsraum gebunden ist, sondern dass sich der Raum auf die von der gerade anwesenden Gruppe gewünschten Parameter einstellt. Dann könnte die Gruppe nicht nur die verwendete Software und Daten (das geht heute schon), sondern ihre Arbeitsumgebung von einem Raum in einen anderen mitnehmen. Derartige, einfach zu nutzende elektronische Gruppenarbeitsräume hätten dann das Potenzial, konventionelle Gruppenarbeitsräume abzulösen.

## 5 Verwandte Arbeiten

Die Idee personalisierbarer Gruppenarbeitsräume baut sowohl auf den Erfahrungen der Autoren als auch auf bisheriger Forschung zu Gruppenarbeitsräumen auf. Die erste Welle der Forschung zu elektronischen Gruppenarbeitsräumen fand Anfang der 90er Jahre statt. Bedeutende Pilotinstallationen entstanden in den USA beispielsweise an der University of Arizona (Nunamaker et al. 1991) und der University of Michigan (Olson et al 1992), in Kanada an der University of Toronto (Buxton 1992, Chattoe et al. 1995) und in Deutschland an der Universität Hohenheim (Lewe&Krcmar 1990, Lewe 1995). In diesen Räumen wurde die Komplexität ihrer Nutzung von den Anwendern auf geschulte Technikspezialisten („Facilitatoren“) verlagert, eine integrierte Techniksteuerung unterblieb weitgehend.

Als Software für die Moderation von Sitzungen und Workshops hat sich weitgehend GroupSystems ([www.groupsystems.com](http://www.groupsystems.com)) durchgesetzt. In GroupSystems steht zwar ein zentrales Agenda-Werkzeug zur Steuerung der Applikationen auf den einzelnen Rechnern zur Verfügung, aber eine Steuerung von Hardwarekomponenten ist nicht vorgesehen. Schon die Einrichtung und Steuerung der Software ist aber so komplex, dass sich GroupSystems nicht in die Breite durchsetzen konnte. Schon früh wurde deshalb überlegt, ob sich die Moderation nicht durch die Verwendung vorgefertigter Schablonen für bestimmte Sitzungstypen vereinfachen ließe (vgl. z.B. Schwabe 1995), denn nur in den Gruppen, in denen die Sitzungstypen standardisiert und damit zur Routine wurden, blieb die Nutzung auch dann erhalten, wenn kein professioneller Moderator mehr Hilfe leisten konnte (Briggs et al. 1998). Eine Standardisierung von Sitzungen erfordert die Identifikation von atomaren Sitzungsaktivitäten, die dann zu generischen Sitzungsmustern zusammengesetzt werden können. Die Media Synchronicity Theorie (Dennis&Valacich 1999, Schwabe 2001) schlägt zwei derartige atomare Kommunikationsprozessstypen vor, der Thinklets-Ansatz (Briggs et al. 2001) gibt insgesamt sieben Kommunikationsprozesse vor (Diverge, converge, organize, elaborate, abstract, evaluate, build consensus).

Von dem in diesem Artikel beschriebenen Nutzungsszenario „Sitzung“ oder „Workshop“ sind Nutzungsszenarien zur Besprechungsunterstützung (vgl. (Engel et al. 2001)) und für die weitgehend unstrukturierte Projektarbeit zu unterscheiden. Sitzungen und Workshops haben typischerweise mehr als 5 Teilnehmer, während bei Besprechungen und unstrukturierter Projektarbeit meist weniger als 5 Personen anwesend sind. Die Hauptkonsequenz liegt in den Anforderungen an die Prozessunterstützung: Während Gruppen ab ca. 5 Personen einer Moderation und damit einer Prozesssteuerung bedürfen, können kleinere Gruppen sich auch ad hoc organisieren. Da größere Gruppen andere Probleme als kleinere haben, sind viele bisherige Ergebnisse aus dem CSCW-Umfeld (für eine Übersicht vgl. Streitz 2001) nur eingeschränkt übertragbar. Diese haben ihre Stärke in der noch weitergehenden Integration von Endgeräten zu einer einheitlichen Arbeitsumgebung (z.B. Verbindung mehrerer physischer elektronischer Tafeln zu einer logischen Tafel).

Unser Ansatz unterscheidet sich auch von „didaktischen Netzwerken“ (vgl. z.B. <http://www.esera.de/index.htm>) - das sind so genannte Lehrer-Schüler-Systeme - bei denen es in erster Linie um eine hardwarenahe Übertragung von Bildschirmhalten geht, die aber beispielsweise über serielle Schnittstellen angebundene Endgeräte wie Kameras nicht steuern können.

## 6 Literaturverzeichnis

- Briggs, R.; Adkins, M.; Mittleman, D.; Kruse, J.; Miller, S.; Nunamaker, J. (1998): A Technology Transition Model Derived From Qualitative Field Investigation of GSS Use Aboard the U.S.S. CORONADO. In: *Journal of Management Information Systems*. Winter, 1998-99, 15(3), 151-193. Abruf im Web vom 29.01.02 <http://www.cmi.arizona.edu/personal/bbriggs/Downloads/ttm.doc>
- Briggs, R.; de Vreede, G.; Nunamaker, J.; Tobey, D. (2001): ThinkLets: Achieving Predictable, Repeatable Patterns of Group Interaction with Group Support Systems (GSS). In: *Proceedings of the 34th Annual Hawaii International Conference on System Sciences (HICSS-34)*, IEEE Press, CD-ROM.
- Buxton, W. (1992): Telepresence: Integrating shared task and person spaces. In: *Proceedings of Graphics Interface '92*, 1992, S. 123-129.
- Chattoe, J.; Leach, P.; Riesenbach, R. (1995): *The Ontario Telepresence Project : Final report*, Information Technology Research Centre, Telecommunications Research Institute of Ontario. Toronto 1995, [http://www.dgp.toronto.edu/tp/techdocs/Final\\_Report.pdf](http://www.dgp.toronto.edu/tp/techdocs/Final_Report.pdf), zugegriffen am 23.06.1998.
- Dennis, A.; Valacich, J. (1999). Rethinking Media Richness: Towards a Theory of Media Synchronicity. In: *Proceedings of the 32<sup>nd</sup> Annual Hawaii International Conference on System Sciences 1999*, CD-ROM, IEEE Computer Society, Los Alamitos, 10 pages.
- Engel, A.; Kaiser, S.; Mayer, A.: Telebesprechung und Telepräsenz. In: Schwabe, G.; Streitz, N.; Unland, R.: *CSCW-Kompodium*, Springer, Heidelberg et al. S. 222-237.
- Lewe, H. (1995): *Computer Aided Team und Produktivität*. Gabler, Wiesbaden.
- Lewe, H.; Krcmar, H. (1990): The CATeam Meeting Room Environment as a Human-Computer Interface. In: Gibbs, S.; Verrijn-Stuart, A.: *Multi-User Interfaces and Applications, Proceedings of the IFIP WG 8.4 Conference on Multi-User Interfaces and Applications in Heraklion, Kreta 24.-26.9.1990*, North-Holland, S. 143-158.
- Nunamaker, J.; Dennis, A.; Valacich, J.; Vogel, D.; George, J. (1991): Electronic meetings to support group work. In: *Communications of the ACM*, Vol. 34, Nr. 7 July, S. 40 - 61.
- Olson, G.; Olson, J.; Kelley, J.; Mack, L.; Cornell, P.; Luchetti, R. (1992): Flexible facilities for electronic meetings. In: Bostrom, R.; Watson, R.; Kinney, S.: *Computer augmented teamwork: A guided tour*. Van Nostrand Reinhold, New York, S. 183-196.
- Schwabe, G.: *Objekte der Gruppenarbeit - ein Konzept für das Computer Aided Team*, Gabler, Wiesbaden 1995.
- Schwabe, G. (2001): Mediensynchronizität - Theorie und Anwendung bei Gruppenarbeit und Lernen. In: Hesse, F.; Friedrich, H.: *Partizipation und Interaktion im virtuellen Seminar*, Waxmann, Münster, S.111-134.
- Schwabe, G.: *(T)Räume der Zusammenarbeit- Probleme und neue Ansätze der elektronischen Sitzungsunterstützung*, Erscheint in: I-COM, Nr. 1 2002.
- Streitz, N. (2001): Kooperative Gebäude und Roomware. In: Schwabe, G.; Streitz, N.; Unland, R.: *CSCW-Kompodium*, Springer, Heidelberg et al. S. 518-534.
- Weiser, M. (1991): The computer of the 21st century. In: *Scientific American, Special Issue on Communications, Computers and Networks*, Vol. 265, Nr. 3 (September), S. 94 - 106.